This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2924910, IEEE Transactions on Circuits and Systems for Video Technology

A Tunable Selective Encryption Scheme for H.265/HEVC Based on Chroma IPM and Coefficient Scrambling

Fei Peng, *IEEE Member*, Xiang Zhang, Zi-Xing Lin, and Min Long

***Abstract*—Designing selective encryption (SE) schemes for H.265/HEVC has been attracted much attention with the advent of H.265/HEVC codec in the past two decades. However, most SE algorithms for H.265/HEVC encrypt the syntax elements in bypass mode to keep the bit rate. Moreover, the edge region of the video data is not sufficiently protected. To produce large visual distortion and edge loss, a tunable SE scheme for H.265/HEVC based on chroma intra prediction mode (IPM) and coefficient scrambling is proposed. Firstly, a pseudo-random number sequence is generated by AES-CTR. Then, the prediction, residual and reconstruction information in H.265/HEVC encoding process are encrypted by a pseudo-random sequence. It encrypts the syntax elements of context-based adaptive binary arithmetic coding (CABAC) in bypass mode. Some syntax elements including chroma IPM in regular mode are encrypted as well. To further protect the edge information, a coefficient scrambling is adopted. The edge information of each frame is extracted and the Transform Units (TUs) are classified according to it. Then, coefficients of the TUs containing edge are scrambled. Finally, a sign used for marking the type of each TU is embedded into a coefficient. Experimental results and analysis show that the proposed scheme has better visual distortion and subjective evaluation results compared with some existing H.265/HEVC SE algorithms. Meanwhile, users can flexibly use the proposed SE scheme according to encryption performance and bit rate requirements, which is attractive in the scenario of protecting video in cloud servers.**

***Index Terms*—H.265/HEVC, Selective encryption, Chroma intra prediction mode, Coefficient scrambling, Edge extraction**

I. INTRODUCTION

WITH the development of big data and multimedia technology, more and more video applications such as video-on-demand (VOD), video conference and video surveillance have been widely used. However, in video transmission and storage, sensitive information in the video may be leaked because insecurity of the public channels and unreliability of the cloud service provider. As a conventional protection means, cryptography is adopted for protecting sensitive data [1-2]. Thus, many researches have been devoted to image and video encryption in the past decade [3-4].

In the early stage, video encryption directly encrypted the coded video as a binary sequence without considering video coding [5-6]. This kind of encryption is called Naive Encryption Algorithm (NEA). NEA is incompatible with video codec, which means that the encrypted video cannot be decoded by standard decoders. Meanwhile, encryption of the video with high resolutions (such as 4K, 8K video) by NEA results in high computational complexity, and it does not meet the requirements of mobile devices and real-time transmission. For these reasons, it is not suitable for practical applications. Therefore, research of video encryption is mainly focused on SE.

SE elaborately considers the video coding characteristics, and it can protect the video content while keep the compatibility of the encrypted video with the existing coding standards [7]. Before 2013, SE is mainly focused on H.264/AVC [8-11]. As H.265/HEVC has significant improvement compared with H.264/AVC in coding efficiency [12], SE for H.265/HEVC received intensive attention in recent years [13-26]. To introduce large distortion and keep bit rate, the syntax elements of CABAC in bypass mode are generally chosen as encryption objects. However, most existing algorithms do not fully considered the relationship between the video quality and the syntax elements, and the information of *I* frames is prone to be disclosed. As *I* frame is of most significance, its encryption will directly affect the encryption performance of *P* frame and *B* frame. Meanwhile, the edge of the video in these encryption schemes is not specifically protected.

In this paper, the relationship between video quality and syntax elements is first analyzed, and then a tunable SE scheme for H.265/HEVC based on chroma IPM and coefficient scrambling is proposed. AES-CTR is utilized to generate a pseudo-random sequence, which is used to encrypt multiple syntax elements containing chroma IPM in *I* frame. In addition, coefficient

This work was supported in part by project supported by National Natural Science Foundation of China (Grant No. 61572182, 61370225), project supported by Hunan Provincial Natural Science Foundation of China (Grant No. 15JJ2007).

Fei Peng is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China. (Corresponding author, e-mail: eepengf@gmail.com).

Xiang Zhang is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China. (e-mail: martin_zx@hnu.edu.cn).

Z. X. Lin is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China. (e-mail: b1410z0225@hnu.edu.cn).

Min Long is with the College of Computer and Communication Engineering, Changsha University of Science and Technology Hunan University, Changsha, 410114, China. (e-mail: caslongm@aliyun.cn).

scrambling is implemented to further enhance the security performance. TUs are classified according to the edge information of each frame. The coefficients of the TUs containing edges are scrambled. Meanwhile the others remain unchanged. Finally, the type of the TU is marked by a sign. It can adapt to various application scenarios depending on different application requirements. The main contributions of this paper include the following points:

• The relationship among H.265/HEVC syntax elements and encryption performance is analyzed, and the importance of H.265/HEVC syntax elements for encryption is discussed. It provides a reference for selecting syntax elements of H.265/HEVC for encryption.

• The effect of the encryption of luma IPM and chroma IPM is theoretically analyzed and experimentally tested. It is found that encrypting both luma and chroma IPMs can further introduce larger distortion than only encrypting luma IPM.

• An efficient coefficient scrambling is proposed to protect the edge of videos. The coefficients of the TUs containing edge regions are scrambled, and it can further significantly enhance the encryption security.

• A tunable SE scheme for H.265/HEVC is proposed, where two encryption strengths are provided to meet different application requirements, and it exhibits the flexibility of the proposed algorithm.

The rest of the paper is organized as follows. The related work on video encryption is introduced in Section II. H.265/HEVC codec and syntax elements for encryption are described in Section III. The proposed SE scheme for H.265/HEVC is depicted in Section IV. Experimental results and analysis are provided in Section V. Some discussions are presented in Section VI. The last section concludes the paper.

II. Related Work

A. Naive Encryption Algorithms for H.265/HEVC

In NEA, a coded video is regarded as a bit stream, and encrypted by some traditional ciphers such as Data Encryption Standard (DES) [27] or Advanced Encryption Standard (AES) [28]. In [5], L.Qiao *et al.* proposed a NEA for MPEG videos. International Data Encryption Algorithm (IDEA) is utilized to generate a pseudo-random sequence, and then each bit of MPEG encoded video is encrypted with it [29]. After that, J.Shah *et al*. proposed to encrypt the bit stream of MPEG video with DES or AES [6]. It can protect each bit of the video, and its security performance is guaranteed by DES or AES. Theoretically, a more secure key stream generation method can be applied to further improve the security of NEA. As NEAs are not specially designed for MPEG video codec, they are also applicable to videos encoded by H.265/HEVC. However, the video encrypted by NEA is not compatible with standard video codecs, and the video post-processions such as transcoding cannot be performed to the encrypted bit stream. Thus, it cannot be used in many real application scenarios. In addition, the data volume of video (such as 4K, 8K video) is generally very large, the delay caused by the high complexity of encryption algorithms makes it impossible to meet the requirements of real-time transmission. For these reasons, SE algorithms are proposed.

B. SE Algorithms for H.265/HEVC

SE algorithms usually select some significant syntax elements in the encoding process for encryption, and the encrypted video can be decoded by the standard decoder. However, the decoded video is seriously distorted, and illegal users cannot obtain useful information through the encrypted video.

After H.265/HEVC codec was released, SE algorithms for H.265/HEVC are successively proposed. G.V.Wallendael *et al*. pioneered a SE algorithm for H.265/HEVC in 2013 [13]. It first analyzes the relationship among some syntax elements (the residual sign, Motion Vector Difference (MVD) sign, Motion Vector Prediction (MVP) index and Motion Vector (MV) reference index) and video quality. Then the encryption of the combinations of these syntax elements is investigated. It is found that the optimal encryption combination is the residual and MVD sign. After that, more syntax elements including Reference Picture Set (RPS), Quantization Parameter (QP), inter frame information, residual information, de-blocking and Sample Adaptive Offset (SAO) parameters are considered for encryption in [14]. Experimental results demonstrate that residual and MVD sign are still the optimal combination with respect to encryption performance. However, the subjective visual distortion of the optimal scheme is insufficient, and the visual information in the video data may be leaked after decoding. From the perspective of encryption efficiency, H. Hofbauer *et al*. proposed to encrypt the sign of non-zero Alternating Current (AC) coefficients [15], and evaluated the video quality degradation with random encryption of different proportions of AC coefficients. It is found that the encryption of 75% AC coefficients can approximately achieve the performance of encrypting 100% coefficients. However, the subjective visual distortion of the video with only encrypting AC coefficients is still weak. To keep the bit rate of the encrypted video, Z.Shahid *et al.* proposed to encrypt CABAC binstrings in [16]. The binarization of each syntax element is first analyzed, and then an encryption method maintaining the length of bin string is designed. It can achieve good security and the video size is unchanged after encryption. M.Farajallah *et al*. proposed a ROI (Region of Interest) encryption scheme for H.265/HEVC based on tile concept [17]. It localizes the ROI area in tiles, and a set of parameters including MVD sign and value, residual sign and value are encrypted by AES-CFB (Cipher Feedback) mode. To avoid propagation of the encryption outside the ROI region caused by inter prediction, the MVs of non-ROI regions are restricted inside the background region. Y.Tew *et al.* proposed a SE algorithm for H.265/HEVC based on transform skip signal [18]. The sign and the value of MVD and AC coefficients are chosen for encryption. Additionally, transform skip signal is also encrypted. However, it can affect the video encoding process and lead to the reduction of compression ratio. F.Peng *et al*. proposed a SE algorithm for H.265/HEVC based on Rossler chaotic system [19]. A pseudo-random sequence is generated by rossler chaotic system and then it is used to encrypt multiple syntax elements. The algorithm can obtain good encryption performance, but the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2924910, IEEE Transactions on Circuits and Systems for Video Technology

TABLE I
THE CHARACTERISTICS OF THE RELATED H.265/HEVC ENCRYPTION ALGORITHMS

Encryption scheme	Format compliance	Cipher technique	Encryption elements	Context modeling change	Bitrate increase
Qiao[5]	No	IDEA	Bit stream	-	No
Jolly[6]	No	DES or AES	Bit stream	-	No
Glen[13]	Yes	AES	MVD sign, Residual sign, RefFrmIdx, MVPIdx	No	No
Glen[14]	Yes	AES	MVD sign and value, Residual sign and value, RPS, Delta QP, RefFrmIdx, Merge index, MVPIdx, SAO parameter	No	No
Heinz[15]	Yes	-	AC coefficients sign	No	No
Shahid[16]	Yes	AES-CFB	MVD sign and value, Residual sign and value	No	No
Farajallah[17]	Yes	AES-CFB	MVD sign and value, Residual sign and value	No	No
Tew[18]	Yes	Hash function	AC coefficients sign, MVD sign	No	No
Peng[19]	Yes	Rossler Chaotic system	MVD sign, MVPIdx, Merge index, RefFrmIdx, Residual sign, DC Coefficient	Yes	Yes
Yang[20]	Yes	AES	MVD sign and value, Residual sign, Delta QP	No	No
Vasileios[21]	Yes	AES-256	Residual Coefficient	No	Yes
Sallam[22]	Yes	chaotic logistic map	MVD sign, DCT coefficient sign	No	No
Sallam[23]	Yes	RC6	MVD sign and value, DCT coefficient sign and value	No	No
Sallam[24]	Yes	RC6	MVD sign and value, DCT coefficient sign and value, Delta QP, SAO parameter, RefFrmIdx, Residual size	No	No
Li[25]	Yes	chaotic logistic map	Residual value	No	No
Boyadjis[26]	Yes	AES-CTR	Luma IPM, MVD sign and value, Residual sign and value, RefFrmIdx, Merge index, SAO parameter, MVPIdx	Yes	Yes

bit rate of the video is increased. M. Yang *et al*. proposed a SE algorithm for H.265/HEVC [20]. The syntax elements including QP offset, the sign and the value of MVD and residual are chosen for encryption. It can achieve good encryption security while keeping the bit rate. As *P* and *B* frames are predicted based on *I* frames, V.A.Memos *et al*. stated that encryption of *I* frames can also achieve good visual distortion [21]. The residual coefficients in *I* frames are chosen for encryption. Although it can improve the encryption efficiency, the encryption space is small and the bit rate is increased. A.I.Sallam *et al*. proposed a SE algorithm using a chaotic logistic map. The chaotic logistic map is used to generate a pseudo-random sequence, and XOR operation is made between the pseudo-random sequence and the sign of MVD and Discrete Cosine Transform (DCT) coefficients [22]. It can achieve low complexity and keep the bit rate, but the encryption space is mall and the distortion of the encrypted video is limited. Consequently, they used RC6 to generate pseudo-random sequence [23]. It can achieve better encryption performance compared with the method in [19] meanwhile keeping the bit rate. After that, they further proposed a SE algorithm for H.265/HEVC using RC6 in different operation modes [24]. More syntax elements such as sign of residual coefficients and MVD, QP offsets, SAO sign, residual size and reference picture index are chosen for encryption. The encryption performance is improved to some extent. Whereas, without encrypting the IPM, the visual information of *I* frames may be leaked. From the perspective of truncated rice (TR) code, J.Li *et al*. proposed a SE algorithm for HEVC [25]. It can keep the parameter *R* of TR code, and the bit rate remains constant during the encryption. However, as it only encrypts the residual sign, the encryption space is not big enough and the distortion of the encrypted video is also insufficient. B.Boyadjis *et al*. proposed a SE algorithm compatible with both H.264/AVC and H.265/HEVC [26], where luma IPM is chosen as encryption object as its great influence on the video quality of the decoded video data. The distortion of *I* frames is improved, but the protection of the edge regions is not considered. Thus, the profile of object in the video still can be recognized.

The characteristics of the above-mentioned H.265/HEVC encryption algorithms are listed in Table I. From Table I, one can see that most SE algorithms choose the syntax elements in bypass mode for encryption to guarantee that the size of the encrypted video is unchanged. However, these syntax elements do not contain chroma IPM, and it results in insufficient distortion of *I* frames, because it is critical in video coding. Meanwhile, the edge protection is not taken into consideration, illegal users may acquire significant information from the edge region. Thus, the security of these SE algorithms needs further improvement.

To solve the fore-mentioned problems in the existing SE algorithms for H.265/HEVC, a tunable SE scheme for H.265/HEVC based on chroma IPM and coefficient scrambling is proposed in this paper.

III. H.265/HEVC CODEC TECHNOLOGY AND SYNTAX ELEMENTS FOR ENCRYPTION

A. The Framework of H.265/HEVC

H.265/HEVC codec preserves the coding framework of H.264/AVC [30]. It is a hybrid coding framework and the complete coding process is shown in Fig.1. It includes prediction (the red dotted box in Fig.1), transform (the blue dotted box) and entropy coding (the green dotted box). Given a frame F_n, it is first divided into several Coding Tree Units (CTUs) with a size of 64×64, and then these CTUs are divided into small Coding Units (CUs) with different sizes (from 8×8 to 64×64). The CUs are further divided into Prediction Units (PUs) with different sizes (from 4×4 to 32×32). After that, intra or inter prediction is applied to obtain prediction values PR_n. Subtract the original values of the current PU with the prediction values PR_n to obtain the residual block D_n. Then, it is divided into TUs with different sizes (from 4×4 to 32×32), which will be further trans-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2924910, IEEE Transactions on Circuits and Systems for Video Technology

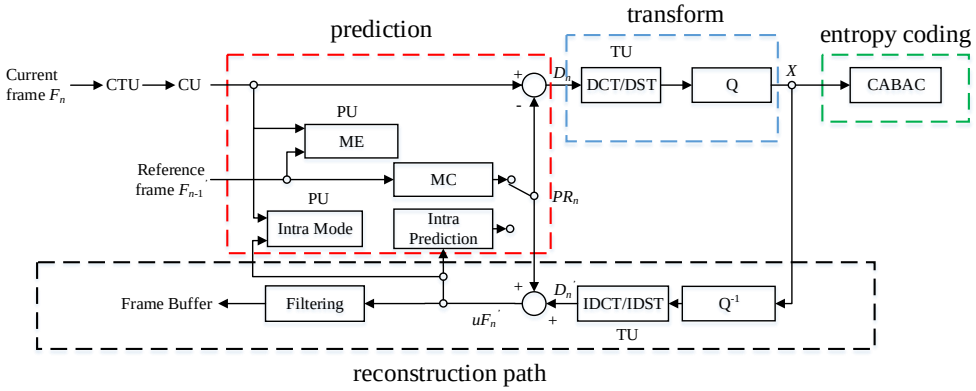

Fig. 1. The framework of H.265/HEVC codec

formed and quantized to residual coefficients *X*. Finally, the residual coefficients form the coded video stream by entropy coding with some side information. Decoding is just an inverse process of the encoding. Moreover, the encoder also has a reconstruction path (the black dotted box), which is used to obtain the reconstruction frame, and it is stored in a buffer. Compared with H.264/HEVC, the prediction, transform, quantization, entropy coding and filtering of H.265/HEVC have been further optimized.

B. *Analysis of Syntax Elements for Encryption*

According to the framework of H.265/HEVC illustrated in Fig.1, the original pixels are represented by the prediction values and the residual values in the encoding process, and it is formulated as

$$F_n = PR_n + D_n, \tag{1}$$

where F_n represents the original pixels of the current frame, PR_n represents the prediction values, and D_n is the residual values.

Conversely, the decoding process can be represented as

$$F_n' = filter_{pa}\left(PR_n + D_n'\right), \tag{2}$$

where F_n' is the decoded pixels of the current frame, D_n' is the reconstruction residuals obtained by inverse quantization and inverse transform, and $filter_{pa}(\cdot)$ is a filtering function with a default parameter *pa*.

During the encoding process, most of the significant syntax elements are used to record PR_n, D_n' and filtering parameters for reconstructing the frame. The main syntax elements of H.265/HEVC are listed in Table II. There are two entropy encoding modes for these syntax elements in CABAC: regular mode and bypass mode [31], and they are shown in Fig.2. If the syntax elements are encoded in regular mode, the bit stream of

TABLE II
THE MAIN SYNTAX ELEMENTS OF H.265/HEVC

Syntax element	Entropy mode	Impact	category
Luma IPM	Regular	I Frames(mainly)	Prediction
Chroma IPM	Regular	I Frames(mainly)	Prediction
Residual sign	Bypass	I,P and B Frames	Residual
Residual value	Bypass	I,P and B Frames	Residual
Delta QP	Bypass	I,P and B Frames	Residual
Merge index	Regular, Bypass	P and B Frames	Prediction
MVD sign	Bypass	P and B Frames	Prediction
MVD value	Bypass	P and B Frames	Prediction
MVPIdx	Regular	P and B Frames	Prediction
RefFrmIdx	Regular, Bypass	P and B Frames	Prediction
SAO parameter	Bypass	I,P and B Frames	Filtering

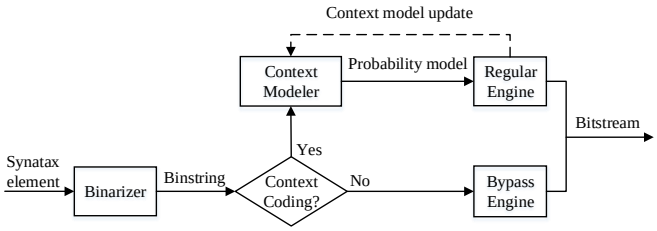

Fig. 2. The diagram of entropy coding in CABAC

these syntax elements is coded based on a probability, which is adaptively updated according to the context. However, the syntax elements in bypass mode are coded with equal probability. Therefore, the encryption of the syntax elements in regular mode will change the video size.

According to (2), the syntax elements in Table II can be further classified into three categories: prediction, residual, and filtering syntax elements.

1) *Prediction Syntax Elements*

Prediction syntax elements are used to record the prediction values, and encryption of them would significantly change the prediction values PR_n. The decoded pixels of the encrypted F_n' is

$$F_n' = filter_{pa}\left(enc\left(PR_n\right) + D_n'\right), \tag{3}$$

where $enc(\cdot)$ is an encryption algorithm. The decoded pixels are significantly different from the original pixels due to the change in the prediction values. Luma IPM, Chroma IPM, Merge index, MVD sign, MVD value, MVPidx, and RefFrmIdx listed in Table II belong to this category. The entropy coding mode of these syntax elements includes regular and bypass mode.

2) *Residual Syntax Elements*

The second category is residual syntax elements. Encryption of these syntax elements will change the residual values D_n'. The decoded pixels of encrypted frame can be represented as

$$F_n' = filter_{pa}\left(PR_n + enc\left(D_n'\right)\right). \tag{4}$$

Similarly, the change of the residual values would also cause significant distortion in the decoded pixels. The syntax elements of this category include Residual sign, Residual value, and Delta QP. The entropy coding mode of these syntax elements is only bypass mode.

Since the decoded pixels are mainly restored by the prediction and residual values, these two categories of syntax elements above can directly determine the decoded pixel values.

3) Filtering Syntax Elements

The third category is filtering parameter. Filtering process is mainly utilized to weaken the blocking and ringing effect in the encoding process. The decoded pixels after encryption of the filtering parameter can be represented as

$$F_n' = filter_{enc(pa)}\left(PR_n + D_n'\right). \quad (5)$$

Actually, encryption of the filtering parameter has limited influence on the quality of the decoded pixels. The syntax element belonging to this category is SAO parameter, and its entropy coding mode is bypass mode.

From the above analysis, except for SAO parameter, encryption of the rest syntax elements listed in Table II can result in large visual distortion in the decoded video frames. As for H.265/HEVC, the rate control is not enabled in the default settings, so the Delta QP is 0. In this paper, except Delta QP, the rest syntax elements listed in Table II are chosen for encryption.

IV. The Proposed Scheme

The proposed scheme is tunable and it includes two parts, they are SE algorithm based on multiple syntax elements (Enc) and coefficients scrambling based on edge extraction (Scr). In fact, Enc can be applied to encrypt videos independently, and it can provide a good encryption performance with small bit rate increment, which can be applied to real-time transmission scenes. To further protect the edge information, an optional coefficient scrambling scheme (Scr) is proposed. It can provide a larger distortion in encrypted frames with an acceptable bit rate increment. Sen (Enc+Scr) is suitable for cloud storage and the videos with high security level. Thus, users can choose to use Enc or Sen (Enc+Scr) depending on the application requirements.

A. SE Algorithm Based on Multiple Syntax Elements (Enc)

From the analysis in Section III.*B*, most syntax elements listed in Table II can affect the decoded frames in various degrees. Therefore, they are encrypted to protect the video content. The diagram of this encryption algorithm is shown in Fig.3.

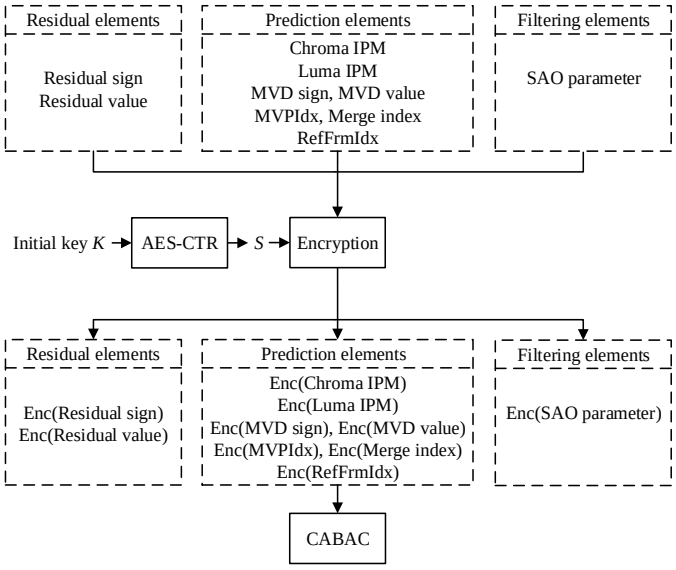

Fig. 3. The diagram of the SE algorithm based on multiple syntax elements

First, a pseudo-random sequence *S* (the length of *S* depends on the encryption elements) is generated by AES-CTR with an initial key *K*, which can be represented as

$$S = \text{AES-CTR}(K), \quad (6)$$

where AES-CTR(·) represents that AES is operated in counter mode. It turns a block cipher into a stream cipher and generates a sequence with arbitrary length by encrypting successive values of a counter. The counter is a sequence produced without repeating for a long time, and an increment-by-one counter is the most popular one. More details of CTR is described in [32].

After generating the pseudo-random sequence *S*, it is used to encrypt each binarized syntax element. As the binarization method of each syntax element is different, the encryption of each syntax element is described in the following sub-sections.

1) Encryption of Luma IPM

Luma IPM plays a key role in *I* frame coding. In [26], B.Boyadjis *et al*. proposed to encrypt luma IPM. The encryption is accomplished by performing XOR operations between the number of candidate mode list or the mode offset with a key stream. Meanwhile, the coefficient scanning mode is synchronized by mapping the position of the last coefficient to solve the decoding failure caused by the encryption of the prediction mode. Here, the luma IPM is directly encrypted. As for H.265/HEVC, there are 35 prediction modes (the number is from 0 to 34) for the luma IPM. However, the number of the luma IPM is not directly recorded during the encoding process, and a candidate mode list with a length of 3 is first established according to the neighboring PUs. If the current luma IPM is in the list, the list number is recorded. The list number *preIdx* is encrypted by the following modular addition.

$$enc\_preIdx = \left(preIdx + S_i\right)\%3 \quad 0 \le S_i \le 3, \quad (7)$$

where S_i represents a part of the pseudo-random sequence *S*. If the current luma IPM is not in the list, it will be compared with the candidate modes in the list, and a 5-bit offset *dir* will be recorded. The encryption is represented as

$$enc\_dir = dir \oplus S_i \quad 0 \le S_i \le 31, \quad (8)$$

where $\oplus$ represents XOR operation.

As the encryption of the luma IPM leads to decoding failure, the encrypted luma IPMs of each PU derived by the encrypted list number or offset are recorded to form a new array in the proposed scheme. They are used to determine the scanning mode instead of the original luma IPMs. It can solve the asynchronous problem between the encrypted luma IPM and the coefficient scanning mode.

2) Encryption of Chroma IPM

In H.265/HEVC codec, there are five chroma IPMs: planar, vertical, horizontal, DC, and corresponding luma IPM, respectively. Since the encryption of the luma IPM can probably affect the chroma components, the chroma IPM is not encrypted in the scheme proposed by B.Boyadjis *et al* [26]. However, according to the chroma IPM encoding, there exist some situations where the chroma IPMs are not affected by the luma component (the related analysis is given in Section VI). So the chroma IPM needs to be further encrypted by

$$enc\_uiDirChroma = uiDirChroma \oplus S_i \quad 0 \le S_i \le 3, \quad (9)$$

where *uiDirChroma* is the list number of the chroma IPM in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2924910, IEEE Transactions on Circuits and Systems for Video Technology

Situation 2. Similarly, the encrypted chroma IPM also needs to be recorded to determine the coefficient scanning mode.

3) Encryption of Residual Sign and Value

In H.265/HEVC codec, the residual sign and value in TUs are separately coded, and the sign of each non-zero coefficient is recorded in bit stream. Assuming that the sign of the coefficient is represented as *coefsign*, *coefsign*=0 represents that the coefficient is positive. Otherwise, the coefficient is negative. The encryption of the residual sign is defined as

$$enc\_coefsign = coefsign \oplus S_i \quad 0 \le S_i \le 1. \tag{10}$$

Meanwhile, a new sign coding method is provided in H.265/HEVC codec. The sum of all non-zero coefficients value in the entire TU is first calculated, and the sign of the last scanned coefficient is represented by the parity of it. If the sign does not correspond to the parity, the value of one coefficient would be adjusted via rate distortion optimization. The corresponding syntax element is *SignHideFlag*. If it is true, the encryption of the last scanned coefficient will be ineffective. Thus, it is recommended to set *SignHideFlag*=false.

In H.265/HEVC codec, the residual value *absCoefLevel* is recorded by the base level and remaining level, and it is represented as

$$absCoefLevel = CoefbaseLevel + CoefremainingLevel, \tag{11}$$

where *CoefbaseLevel* represents the base level. Since it is encoded in regular mode, it is not used for encryption. *CoefremainingLevel* is the remaining level, and it is encoded in bypass mode and binarized through TR code and *K*-order Exp-Golomb code. It consists of a prefix and suffix. The prefix represents the level's interval and the suffix is the offset in the interval. Encryption of the suffix *Coefsuffix* can change the coefficient value without bit rate increment. The encryption is represented by

$$enc\_Coefsuffix = Coefsuffix \oplus S_i, \tag{12}$$

where S_i is changed according to the value of *CoefremainingLevel*. Moreover, the change of *CoefremainingLevel* may update the parameter *RiceParam*, affecting the coding of the next coefficient. To avoid decoding failure, it is necessary to record the current encrypted coefficient and transmit it to the subsequent coding. The encrypted coefficient is formulated as

$$\begin{aligned} enc\_absCoefLevel = {} & CoefbaseLevel + \\ & CoefremainingLevel + enc\_Coefsuffix - Coefsuffix \end{aligned}. \tag{13}$$

4) Encryption of MVD Sign and Value

MVD is similar to residual, and its sign and value are separately encoded. Furthermore, MVD has a horizontal component *MVDHor* and a vertical component *MVDVer*. Their signs are encrypted as

$$enc\_MVDHorsign = MVDHorsign \oplus S_i \quad 0 \le S_i \le 1, \tag{14}$$

$$enc\_MVDVersign = MVDVersign \oplus S_i \quad 0 \le S_i \le 1, \tag{15}$$

where *MVDHorsign* is the sign of the horizontal component and *MVDVersign* is the sign of the vertical component.

MVD value is also represented by the base level and the remaining level. The base level is coded in regular mode and it is not encrypted in the algorithm. The remaining level is binarized by 1-order Exp-Golomb code. Similar to the residual, the suffix *MVDsuffix* is encrypted as

$$enc\_MVDsuffix = MVDsuffix \oplus S_i, \tag{16}$$

where S_i is changed according to the remaining level of MVD. After the sign and value are encrypted, the horizontal and vertical component are exchanged to further enhance the encryption performance, and it is represented as

$$(MVDVer, MVDHor) = Exch(MVDHor, MVDVer) \quad if \ \ S_i = 1, \tag{17}$$

where $Exch(\cdot)$ is a function that exchanges the horizontal and the vertical component.

5) Encryption of Merge Index

The merge mode is used for inter prediction. In this mode, a candidate MV reference list with a length of 5 is directly established by the neighboring PUs, and then an optimal MV in the list is chosen by rate distortion optimization as the MV of the current PU. Its number in the list is Merge index, and Merge index varies from 0 to 4. So modular addition is utilized to encrypt Merge index *uiUnaryIdx*, and it is defined as

$$enc\_uiUnaryIdx = (uiUnaryIdx + S_i)\%5 \quad 0 \le S_i \le 3. \tag{18}$$

6) Encryption of MVP Index

In H.265/HEVC codec, except for Merge mode, there exist another inter prediction mode called as AMVP (Advanced Motion Vector Prediction). A candidate MV list is created in this mode, and the length is 2. The optimal prediction MV is first chosen from the candidate list, and then MVD is obtained by subtracting the optimal prediction MV from the MV of the current PU. The number of the optimal prediction MV is MVP index *iSymbol*. The encryption of *iSymbol* is defined as

$$enc\_iSymbol = iSymbol \oplus S_i \quad 0 \le S_i \le 1. \tag{19}$$

7) Encryption of Reference Frame Index

The reference frame index is the number of reference frame corresponding to the current MV. In the encoding process, the upper bound varies according to the reference frame buffer. The encryption makes sense when the maximum number of the current frame in the buffer is greater than 1. Therefore, the encryption of the reference frame index *RFI* is

$$enc\_RFI = \begin{cases} RFI \oplus S_i \quad 0 \le S_i \le 1 \ \ if \ \ RN = 2 \\ (RFI + S_i)\%3 \quad 0 \le S_i \le 1 \ \ if \ \ RN = 3 \\ RFI \oplus S_i \quad 0 \le S_i \le 3 \ \ if \ \ RN = 4 \end{cases}, \tag{20}$$

where *RN* is the current number of the frames in the buffer, and its maximal value is set to 4 in this paper.

8) Encryption of SAO Parameter

SAO is a novel technique in H.265/HEVC, and used to weaken the ringing effect [33]. The pixels in a CTU are compensated by SAO. There are two compensation methods in SAO, and they are Edge Offset (EO) and Band Offset (BO). There are 4 modes in EO and 32 bands in BO. The EO mode *typeIdc* and the BO type *typeAuxInfo* are encoded in bypass mode. The encryption can be represented as

$$enc\_typeIdc = typeIdc \oplus S_i \quad 0 \le S_i \le 3, \tag{21}$$

$$enc\_typeAuxInfo = typeAuxInfo \oplus S_i \quad 0 \le S_i \le 31. \tag{22}$$

For the decryption of these encrypted syntax elements, it is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2924910, IEEE Transactions on Circuits and Systems for Video Technology

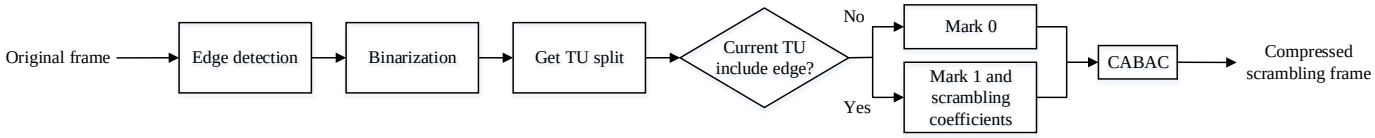

Fig. 4. The diagram of the coefficient scrambling

TABLE III
THE PSEUDO CODE OF MAP GENERATION FUNCTION

Algorithm Pseudo-Code for map generation

Input: Number of non-zero coefficients: Num_{non}
Output: Scrambling Mapping: *map*
Begin:
srand(Num_{non})
  for *num*1 ← 0 to Num_{non}-1 **do**
    map[*num*1] = *num*1
  end for
  for *num*2 ← 0 to Num_{non}-2 **do**
    temp = *rand*() % (Num_{non}-*num*2)
    *temp*2 = *map*[*temp*]
    map[*temp*] =*map*[Num_{non}-*num*2-1]
    map[Num_{non}-*num*2-1] = *temp*2
  end for
Return *map*
end

just a reverse of their encryption processes. The original syntax elements can be decrypted by performing the same operations as the encryption process to the encrypted syntax elements.

B. Coefficients Scrambling Based on Edge Extraction (Scr)

The edge of the object may leak video content. Except for encrypting the syntax elements in Section IV.*A*, an optional coefficient scrambling combined with edge extraction is also proposed here to further perturb the residual information. Its diagram is shown in Fig.4. Since the significant information in a video is almost exposed by the edge regions, therefore, after extracting the edge of the original frames, the TUs can be classified according to it. If the current TU contains edges, its coefficients are scrambled and then the TU is marked by "1". Otherwise, the TU is marked by "0" without scrambling. The details are described as follows.

Step 1. The luma component of the current frame F_n is first obtained, and the edge information F_n^{edg} of the luma component is extracted by an edge extraction operator

$$F_n^{edg} = edge(getPicY(F_n)), \tag{23}$$

where *edge*(·) is an edge extraction operator, and *getPicY*(·) represents a method of getting the luma component.

Step 2. F_n^{bin} is obtained by image binarization of the luma edge information

$$F_n^{bin} = imBinary\left(F_n^{edg}\right), \tag{24}$$

where *imBinary*(·) is an image binarization function.

Step 3. Classify the current transform unit *cur_TU* of the frame according to the binary edge image F_n^{bin}

$$cur\_TU = \begin{cases} edg\_TU & if \ \sum pix_{cur\_TU}(x,y) \neq 0 \\ nonedg\_TU & if \ \sum pix_{cur\_TU}(x,y) = 0 \end{cases}, \tag{25}$$

where *edg_TU* represents a TU containing the edge information. On the contrary, *nonedg_TU* represents a TU without edge information, and $pix_{cur\_TU}(x, y)$ is a pixel of F_n^{bin} at the position (x, y) in *cur_TU*.

Step 4. Scramble the coefficients in *edg_TU* as

$$scr\_pcCoef[map[k]] = pcCoef[k] \ \ if \ \ Num_{non} > 1, \tag{26}$$

where Num_{non} represents the number of non-zero coefficients in *edg_TU*, *pcCoef*[*k*] is the k^{th} original non-zero coefficient in *edg_TU*, and *scr_pcCoef*[*k*] is the coefficient after scrambling. *map* is an array which stores the scrambling mapping, and it is generated by a scrambling function

$$map = GenRandMap(Num_{non}), \tag{27}$$

where *GenRandMap*(·) is a scrambling generation function. The pseudo code of this function is listed in Table III. Here, *srand*(·) is used to initialize a pseudo-random number generator, *rand*(·) is used to generate a pseudo-random number, *temp* and *temp*2 are intermediate variables.

Step 5. The sign is reversibly embedded into the last non-zero scanned coefficient *pcCoef*[*last*] in *cur_TU* as

$$pcCoef[last]' = \begin{cases} 2\times pcCoef[last] - wk & if \ \ pcCoef[last] > 0 \\ 2\times pcCoef[last] + wk & if \ \ pcCoef[last] < 0 \end{cases}', \tag{28}$$

where *wk* is a value used for recording the type of TU. Set *wk*=1 if *cur_TU* is *edg_TU*. Otherwise, set *wk*=0.

Step 6. Repeat the above steps to scramble the coefficients of all *edg_TUs* in each frame.

As for the inverse scrambling, it first gets the sign *wk* of the current TU from the embedded last non-zero scanned coefficient *pcCoef*[*last*]*'*

$$wk = abs\left(pcCoef[last]'\%2\right), \tag{29}$$

where *abs*(·) represents absolute value function. The original last non-zero scanned coefficient is recovered as

$$pcCoef[last] = \begin{cases} (pcCoef[last]' + wk)/2 & if \ \ pcCoef[last]' > 0 \\ (pcCoef[last]' - wk)/2 & if \ \ pcCoef[last]' < 0 \end{cases}. \tag{30}$$

If *wk*=1, *cur_TU* is *edg_TU*, and inverse scrambling is performed on the coefficients by *map*.

TABLE IV
VIDEO SEQUENCES AND PARAMETERS

Video sequence	Resolution	Frame Rate
Mobile	352×288	60 fps
Foreman	352×288	
BQSquare	416×240	
BasketballPass	416×240	
BQMall	832×480	
PartyScene	832×480	
Johnny	1280×720	
FourPeople	1280×720	
BasketballDrive	1920×1080	
Kimono	1920×1080	
Traffic	2560×1600	
PeopleOnStreet	2560×1600	

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2924910, IEEE Transactions on Circuits and Systems for Video Technology

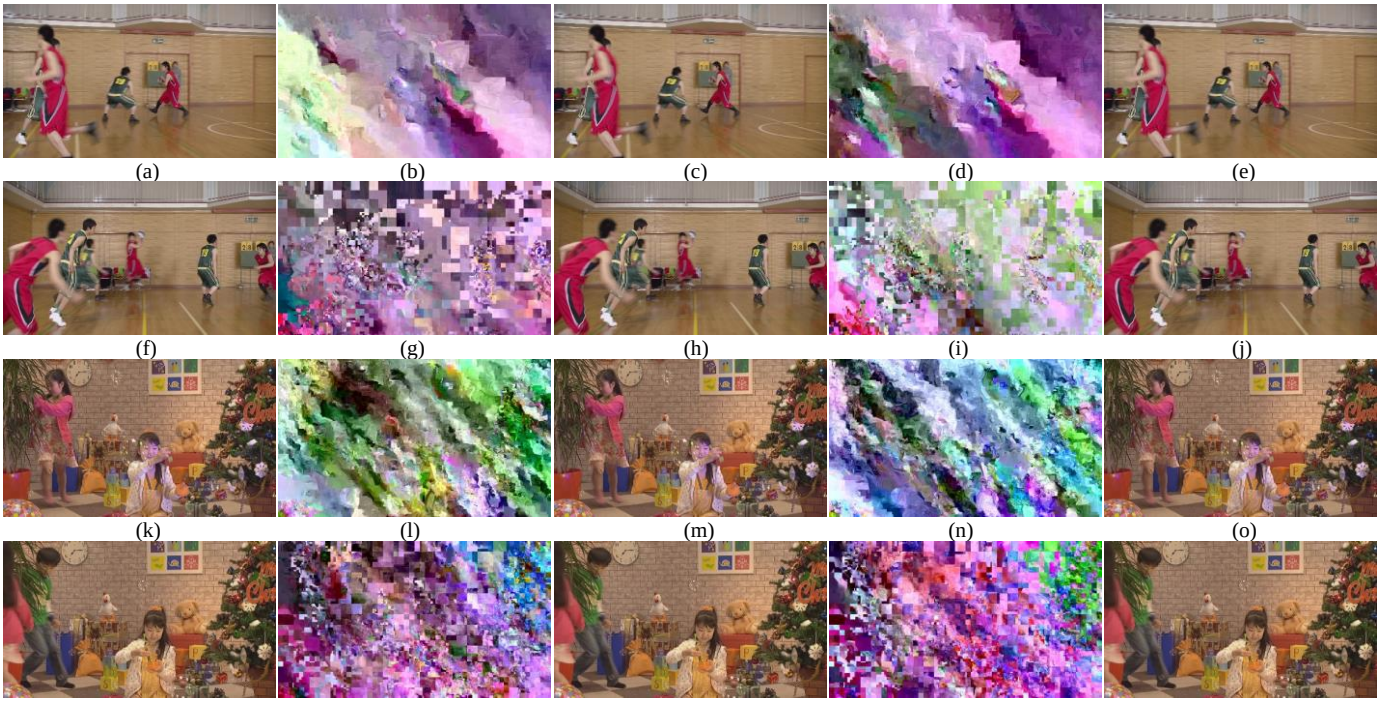

Fig. 5. Experimental results for BasketballDrive (#1 and #50 frame) and PartyScene (#1 and #50 frame). From the first column to the fifth column are the original frames, encrypted frames with Enc, decrypted frames with Enc, encrypted frames with Sen and decrypted frames with Sen

V. Experimental Results and Analysis

A large number of experiments were performed by a personal computer with a configuration: Intel (R) Core (TM) i7-6700HQ CPU @ 2.60GHz, 16GB memory, Windows 10, Microsoft Visual Studio 2010, MATLAB 2016a, and Opencv 2.4.7. The reference software model HM 16.9 is implemented for the proposed scheme. The profile used in HM 16.9 is encoder_lowdelay_main, and Group of Picture (GOP) is 4 (*I* frame followed by three *B* frames). QP is set as 8, 24 and 40, respectively. The edge extraction operator is Sobel [34], and the binarization method is OTSU [35]. 12 video sequences with resolutions from 352×288 to 2560×1600 are chosen for the experiments, which are listed in Table IV. Enc and Sen proposed in this paper are compared with three recent H.265/HEVC SE algorithms including Glenn [14], Sallam [23] and Boyajis [26]. The original videos (only encoding) are used as the reference. AES-CTR is utilized to generate the pseudo-random sequence *S* with an initial key of {0x01, 0x09, 0x09, 0x01, 0x01, 0x02, 0x02, 0x08, 0x01, 0x09, 0x09, 0x04, 0x00, 0x03, 0x01, 0x06} for all algorithms. Cipher block size is 128 bits. An increment-

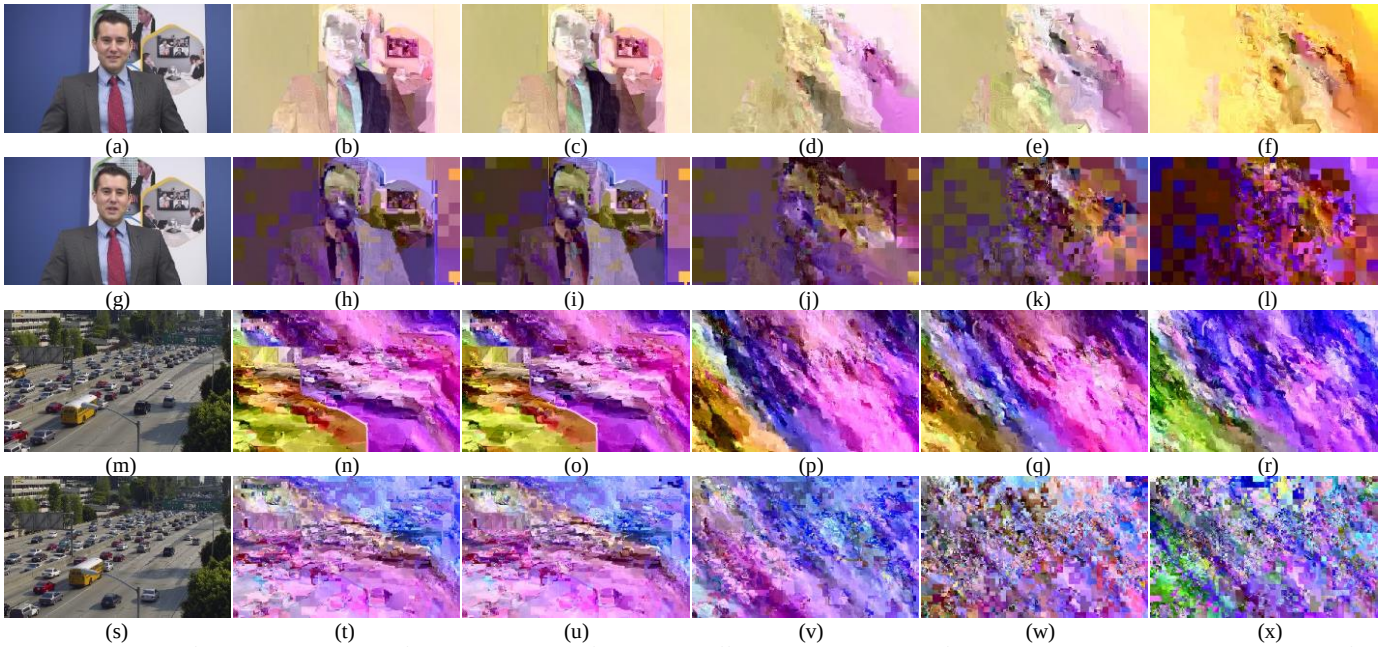

Fig. 6. Comparison of the encryption results of Johnny (#1 and #50 frame) and Traffic sequence (#1 and #50 frame) with 5 encryption algorithms. From the first column to the sixth column are the original frames, encrypted frames with Glenn [14], encrypted frames with Sallam [23], encrypted frames with Boyajis [26], encrypted frames with Enc, and encrypted frames with Sen

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2924910, IEEE Transactions on Circuits and Systems for Video Technology

by-one counter is used and its initial value is set as {0xff, 0xe1, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f}. *SignHideFlag* is set as "false". As there are fluctuations in the encryption performance of a single frame and the number of frames varies for each sequence, it is unified to use 50 frames to test the performance in our experiments for comparison. Here, the optimal scheme (residual and MVD sign are chosen for encryption) by Glenn in [14] is implemented. Residual sign and value, MVD sign and value are used for encryption by Sallam in [23]. luma IPM and other syntax elements in bypass mode are selected for encryption by Boyajis in [26].

A. Experiment Results

To illustrate the validity of Enc and Sen, encryption (QP=24) and decryption are operated on sequences listed in Table IV. The encryption and decryption results of some sample frames are shown in Fig.5. The results indicate the validity of the proposed scheme.

B. Performance Analysis

1) Analysis of Subjective Vision

The objective of SE is to significantly reduce the readability of the video by encrypting partial syntax elements. Therefore, encryption performance is measured by various indicators. Subjective vision is first analyzed in this section. Here, Johnny and Traffic sequences are chosen for analysis, because the texture of Johnny is simple while that of Traffic is relatively complex. Meanwhile, Johnny is static and Traffic is relatively dynamic, so they can represent different kinds of video sequences to some extent. Fig.6 depicts the #1 and #50 decoded frames of two sequences after encryption of 5 algorithms, where QP=24. As seen from Fig.6, the visual distortion of the decoded frame after encryption becomes serious from the left to the right in each row. The performance of Glenn' scheme in [14] and Sallam's scheme in [23] are relatively close in both *I* and *B* frames. However, their encryption performance on low texture videos are not satisfactory. Since Enc, Sen and Boyajis [26] all encrypt IPM, the encryption performance of *I* and *B* frames is significantly improved. Therefore, it can be observed that the encryption of IPM has significant impact on *I* frames and the subsequent inter prediction frames. Furthermore, it can be found that the visual distortion of *B* frames is more serious in Enc and Sen. Especially, the mosaic effect will be more obvious with the increase of the number of encoded frames. The reason is that the proposed scheme scrambles MVD and residual information, while H.265/HEVC codec is encoded based on blocks. The errors caused by encryption is continuously drifted by inter prediction.

TABLE V
THE AVERAGE *PSNR* AND *SSIM* OF FIFTY FRAMES IN FIVE ENCRYPTION ALGORITHMS WITH DIFFERENT QP

Video	QP	*PSNR*						*SSIM*					
		original	Glenn [14]	Sallam [23]	Boyajis [26]	Enc	Sen	original	Glenn [14]	Sallam [23]	Boyajis [26]	Enc	Sen
Mobile	8	51.51	10.68	10.63	10.81	10.89	**10.60**	0.998	0.083	0.083	0.070	0.058	**0.057**
	24	38.92	10.65	10.72	10.64	**10.53**	10.68	0.983	0.107	0.105	0.077	0.076	**0.068**
	40	29.36	11.57	11.49	11.46	10.59	**10.19**	0.861	0.146	0.144	0.110	0.097	**0.087**
Foreman	8	51.42	12.90	12.86	12.43	**12.39**	13.26	0.996	0.270	0.265	0.255	0.222	**0.210**
	24	42.20	12.99	12.90	12.81	12.87	**12.26**	0.960	0.312	0.313	0.320	0.288	**0.248**
	40	33.87	13.01	13.03	**12.40**	12.63	13.24	0.842	0.353	0.352	0.332	**0.323**	0.347
BQSquare	8	51.66	12.86	12.77	12.54	**12.34**	12.55	0.997	0.176	0.171	0.135	0.118	**0.113**
	24	40.50	12.51	12.54	12.81	12.47	**12.27**	0.968	0.211	0.213	0.181	0.170	**0.157**
	40	31.61	12.87	12.90	13.04	12.76	**12.44**	0.842	0.280	0.280	0.238	0.212	**0.206**
BasketballPass	8	51.97	15.23	14.80	15.21	14.89	**13.42**	0.996	0.324	0.315	0.320	0.260	**0.236**
	24	42.28	15.78	15.85	15.48	15.40	**14.04**	0.968	0.413	0.413	0.408	0.372	**0.334**
	40	33.48	16.22	16.23	16.39	16.94	**14.71**	0.808	0.480	0.481	0.457	0.459	**0.432**
BQMall	8	51.39	13.66	13.63	13.96	14.56	**13.59**	0.997	0.239	0.236	0.238	0.215	**0.193**
	24	41.11	14.34	14.33	14.32	14.54	**13.23**	0.965	0.307	0.304	0.301	0.282	**0.259**
	40	32.94	14.59	14.59	14.82	14.40	**13.79**	0.841	0.338	0.338	0.332	0.312	**0.285**
PartyScene	8	51.55	13.10	12.90	13.08	**11.97**	12.64	0.998	0.146	0.143	0.129	0.112	**0.102**
	24	39.20	13.22	13.23	13.23	13.08	**12.24**	0.966	0.162	0.158	0.149	0.133	**0.118**
	40	29.98	13.25	13.27	13.04	12.93	**12.42**	0.758	0.198	0.197	0.189	0.176	**0.158**
Johnny	8	51.70	13.48	13.25	13.38	13.87	**11.62**	0.994	0.489	0.475	0.468	0.449	**0.373**
	24	45.27	13.53	13.36	13.77	13.68	**11.38**	0.968	0.584	0.577	0.569	0.552	**0.456**
	40	38.94	13.02	13.05	13.41	13.40	**11.69**	0.920	0.581	0.588	0.592	0.580	**0.440**
FourPeople	8	51.64	13.34	13.19	**12.76**	13.13	12.87	0.995	0.326	0.326	0.325	0.295	**0.286**
	24	44.67	13.86	13.86	13.61	13.55	**13.13**	0.974	0.441	0.436	0.402	0.381	**0.359**
	40	37.26	13.69	13.73	13.07	12.83	**12.50**	0.917	0.447	0.448	0.420	0.389	**0.380**
BasketballDrive	8	51.30	14.02	**13.96**	14.65	15.21	14.75	0.996	0.480	0.477	0.492	0.496	**0.456**
	24	42.07	13.89	**13.73**	14.72	15.00	14.57	0.929	0.498	0.495	0.545	0.509	**0.470**
	40	36.58	14.62	14.56	15.49	15.26	**14.31**	0.864	0.557	0.555	0.582	0.560	**0.505**
Kimono	8	51.46	11.93	11.83	12.77	**11.71**	12.77	0.995	0.394	0.388	0.395	0.340	**0.297**
	24	42.77	13.17	13.18	12.84	**12.82**	13.05	0.957	0.433	0.429	0.418	0.374	**0.313**
	40	36.62	13.01	13.00	**12.59**	12.97	12.64	0.888	0.462	0.463	0.437	0.432	**0.396**
Traffic	8	51.28	12.46	12.42	12.31	12.53	**12.18**	0.996	0.266	0.262	0.260	0.240	**0.215**
	24	41.70	12.58	12.62	12.81	12.70	**12.25**	0.973	0.354	0.353	0.348	0.328	**0.299**
	40	35.05	13.26	13.26	13.52	13.34	**12.94**	0.874	0.385	0.385	0.372	0.361	**0.340**
PeopleOnStreet	8	51.48	12.79	12.72	12.96	13.12	**12.51**	0.997	0.269	0.264	0.250	0.232	**0.201**
	24	42.16	13.10	13.05	13.23	13.10	**12.49**	0.964	0.314	0.311	0.294	0.262	**0.228**
	40	33.96	12.89	12.95	13.09	13.23	**12.18**	0.856	0.345	0.345	0.332	0.312	**0.280**

2) Analysis of Objective Evaluation Index

To verify the analysis in Section V.*B.1)*, Peak Signal to Noise Ratio (*PSNR*) [36] and Structural Similarity (*SSIM*) [37] are introduced to measure the performance of 5 encryption algorithms. *PSNR* and *SSIM* can be used for measuring the similarity between two frames. The smaller the values of the two indicators are, the higher the distortion of the frame is, which also means a better encryption performance from the perspective of objective visual index. Table V shows the average *PSNR* and *SSIM* of 5 encryption algorithms on 12 videos with different QPs, where the optimal results in the table are in bold and suboptimal results are marked with underline, and the results in the following tables of the other indicators are also highlighted in the same way. From Table V, one can see that the results of most sequences are sequentially reduced from the left to the right, which further prove the analysis results of subjective vision in Section V.B.*1)*. The results of Glenn's scheme [14] and Sallam's scheme [23] are very similar, because the syntax elements for encryption are both less. Although residual and MVD value are additionally encrypted in Sallam's scheme [23], these syntax elements only exist in large coefficients and MVDs, so the encryption space is small. Nevertheless, as more key syntax elements and scrambling are utilized, the *PSNR* and *SSIM* of Enc and Sen are both better than Glenn's scheme [14], Sallam's scheme [23] and Boyajis's scheme [26], In particular, the performance of Enc outperforms those of Glenn [14], Sallam [23] and Boyajis [26] in most cases, and Sen can overall achieve the best results.

According to Table V, the increase of QP reduces the encryption performance in all algorithms, but it has little effect on the comparison of them. Therefore, the average results of three different QPs is used in the subsequent analysis.

3) Analysis of the security of key stream

In the experiments, the pseudo-random stream is generated by AES-CTR for all algorithms. AES is a block cipher established by the U.S. National Institute of Standards and Technology (NIST) in 2001, which is based on a design principle known as a substitution–permutation network. The design of all key lengths of the AES algorithm are sufficient to protect classified information up to the secret level. Protecting information with higher secret level needs to use keys with a bit length of 192 or 256, which is proven to be secure [38], so the security of the pseudo-random stream can be guaranteed. As the focus of this paper is to discuss the influence of different syntax elements in H.265/HEVC on the encryption performance, different key stream generation methods are not discussed and tested in details in this paper. Furthermore, other ciphers such as RC6 [39] also can be used to generate key stream for the proposed scheme.

4) Analysis of Edge Detection

The edge information of the encrypted frame is an important index to measure the performance of video encryption algorithms. Encryption should cause large distortion at the edge region, and the edge information of the video cannot be recognized in the encrypted frame. The edge images of Johnny sequence (#1 and #50 decoded frame) which are extracted by Sobel operator in 5 encryption algorithms with QP=24 are shown in Fig.7, respectively. From Fig.7, one can see that the edge of the encrypted frames in Glenn's scheme [14] and Sallam's scheme [23] are relatively clear, and the people in the frame can be obviously recognized. However, the edge distortion of Boyajis's scheme [26], Enc and Sen is large, especially the edge loss of Sen. The reason is that the coefficient scrambling of the edge regions is applied in Sen. In addition, it can be found from Fig.7 that the edge distortion in *B* frames of Enc and Sen is particularly large, and the encrypted frame cannot be associated

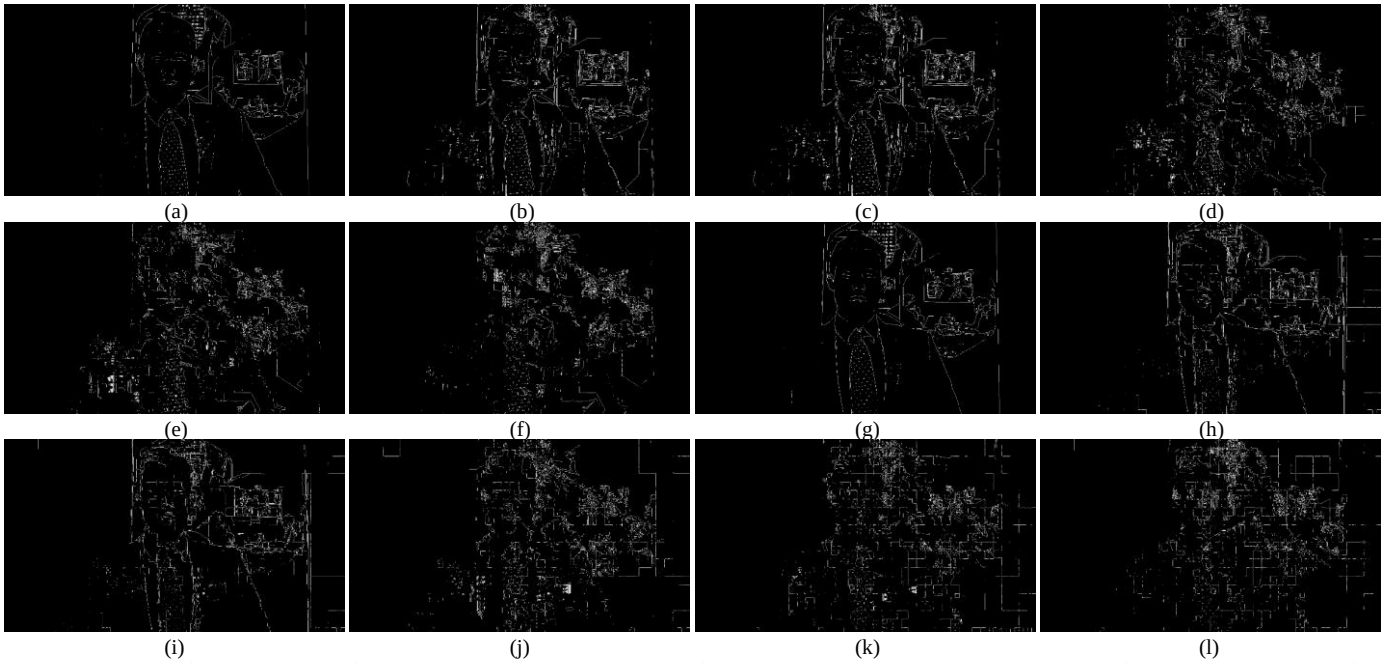

Fig. 7. Comparison of the edge images of Johnny sequence (#1 and #50 decoded frame) with five encryption algorithms (a) edge of the original #1 (b) edge of the encrypted #1 with Glenn [14] (c) edge of the encrypted #1 with Sallam [23] (d) edge of the encrypted #1 with Boyajis [26] (e) edge of the encrypted #1 with Enc (f) edge of the encrypted #1 with Sen (g) edge of the original #50 (h) edge of the encrypted #50 with Glenn [14] (i) edge of the encrypted #50 with Sallam [23] (j) edge of the encrypted #50 with Boyajis [26] (k) edge of the encrypted #50 with Enc (l) edge of the encrypted #50 with Sen

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2924910, IEEE Transactions on Circuits and Systems for Video Technology

with the original frame. This is due to the scrambling of MVD components in (17). Here, an objective index Edge Differential Ratio (*EDR*) defined in [23] is used to evaluate the edge distortion. Generally, the higher the *EDR* is, the larger the edge distortion is. It is calculated by

$$EDR = \frac{\sum_{m=1}^{M} \left| PI(m) - PI_e(m) \right|}{\sum_{m=1}^{M} \left| PI(m) + PI_e(m) \right|}, \quad (31)$$

where M represents the total number of the edge pixels, and $PI(m)$, $PI_e(m)$ represent the m^{th} pixel of the original and the encrypted frames in the edge region, respectively.

The average values of *EDR* are listed in Table VI. From Table VI, Glenn's scheme [14] and Sallam's scheme [23] achieve the similar results in *EDR*, while Boyajis's scheme [26] is better than them. For Enc and Sen, they outperform the other 3 algorithms. Sen can achieve the optimal *EDR* in 9 video sequences and Enc is suboptimal in the corresponding 9 video sequences, whereas Enc can achieve the optimal *EDR* in the rest 3 video sequences and Sen is suboptimal in the corresponding 3 video sequences.

5) Analysis of Information Entropy

Information entropy is a measurement for information certainty, and the information entropy of a frame represents the statistical distribution of all pixels. Larger entropy means higher randomness of pixel distribution, so it can reflect the security of an encryption algorithm to some extent. Information entropy $H(I)$ is defined in [40] as

$$H(I) = \sum_{j=0}^{2^L-1} p(I_j) \log_2 \frac{1}{p(I_j)}, \quad (32)$$

where L represents the length of the pixel value in binary form, and $p(I_j)$ is the occurrence probability of the pixel value I_j.

In theory, the maximum information entropy of an image represented in 8 bits is 8. If the information entropy of an encrypted image is close to 8, it means that a good encryption performance is obtained. The testing results are listed in Table VII. As shown in Table VII, most results of 5 encryption algorithms exceed 7. Sen can achieve the optimal information entropy in 9 video sequences and suboptimal ones in the rest 3 video sequences, and Enc can achieve the optimal information entropy in the rest 3 video sequences and suboptimal ones in 6 video sequences, so the results of Enc and Sen are both better than Glenn's scheme [14], Sallam's scheme [23] and Boyajis's scheme [26]. It indicates that Enc and Sen can achieve good randomness of pixel distribution from the perspective of information entropy.

TABLE VI
THE AVERAGE *EDR* OF FIVE ENCRYPTION ALGORITHMS

Video	Algorithm					
	original	Glenn [14]	Sallam [23]	Boyajis [26]	Enc	Sen
Mobile	0.086	0.878	0.879	0.903	0.912	**0.919**
Foreman	0.134	0.865	0.869	0.917	0.927	**0.933**
BQSquare	0.071	0.817	0.817	0.862	0.875	**0.880**
BasketballPass	0.162	0.837	0.838	0.879	0.892	**0.895**
BQMall	0.136	0.838	0.838	0.884	0.899	**0.907**
PartyScene	0.128	0.859	0.859	0.890	0.903	**0.909**
Johnny	0.182	0.851	0.852	0.901	0.912	**0.915**
FourPeople	0.144	0.830	0.829	0.875	**0.891**	0.888
BasketballDrive	0.153	0.865	0.869	0.922	0.933	**0.938**
Kimono	0.462	0.945	0.945	0.955	**0.961**	0.955
Traffic	0.190	0.881	0.881	0.911	**0.919**	0.917
PeopleOnStreet	0.186	0.882	0.883	0.919	0.926	**0.927**

TABLE VII
THE AVERAGE INFORMATION ENTROPY OF FIVE ENCRYPTION ALGORITHMS

Video	Algorithm					
	original	Glenn [14]	Sallam [23]	Boyajis [26]	Enc	Sen
Mobile	7.568	7.757	7.762	7.812	7.817	**7.822**
Foreman	7.195	7.602	7.599	7.510	**7.641**	7.623
BQSquare	6.744	7.702	7.701	7.728	**7.793**	7.786
BasketballPass	6.544	7.247	7.276	7.323	7.382	**7.494**
BQMall	7.467	7.670	7.681	7.610	7.612	**7.699**
PartyScene	7.411	7.793	7.791	7.793	7.807	**7.839**
Johnny	6.824	7.042	7.088	6.968	7.145	**7.282**
FourPeople	7.350	7.529	7.548	7.508	7.642	**7.710**
BasketballDrive	6.932	7.475	7.483	7.358	7.440	**7.598**
Kimono	6.706	7.605	7.609	7.607	**7.694**	7.692
Traffic	7.348	7.735	7.744	7.711	7.770	**7.801**
PeopleOnStreet	7.122	7.851	7.853	7.822	7.805	**7.869**

TABLE VIII
THE AVERAGE *NPCR* AND *UACI* OF FIVE ENCRYPTION ALGORITHMS

Video	*NPCR*						*UACI*					
	original	Glenn [14]	Sallam [23]	Boyajis [26]	Enc	Sen	original	Glenn [14]	Sallam [23]	Boyajis [26]	Enc	Sen
Mobile	0.72235	0.99559	0.99562	0.99569	0.99581	**0.99600**	0.01533	0.28636	0.28667	0.29201	0.29161	**0.29871**
Foreman	0.70214	0.99569	0.99565	**0.99592**	0.99561	0.99539	0.00982	**0.29351**	0.29340	0.29130	0.28894	0.26443
BQSquare	0.72411	0.99623	**0.99608**	0.99639	0.99639	0.99632	0.01279	0.31411	0.31258	0.30916	**0.32073**	0.31412
BasketballPass	0.70430	0.99073	0.99086	0.99153	0.99171	**0.99334**	0.00950	0.16639	0.16899	0.17023	0.17196	**0.18919**
BQMall	0.70802	0.99436	0.99432	0.99409	0.99422	**0.99456**	0.01062	0.24191	0.24141	0.23522	0.23384	**0.24217**
PartyScene	0.72662	0.99510	0.99515	0.99498	0.99517	**0.99526**	0.01509	0.24876	0.24931	0.24495	0.24883	**0.25371**
Johnny	0.65342	0.99419	0.99384	**0.99386**	0.99342	**0.99737**	0.00574	0.26602	0.26903	0.26239	0.25213	**0.31880**
FourPeople	0.67078	0.99516	0.99541	**0.99615**	0.99632	0.99589	0.00680	0.27614	0.27748	0.29102	**0.29340**	0.29171
BasketballDrive	0.69969	0.99509	**0.99520**	0.99417	0.99403	0.99489	0.00814	0.23606	**0.23898**	0.21722	0.21233	0.22885
Kimono	0.69468	**0.99602**	0.99598	0.99569	0.99655	0.99625	0.00776	0.31098	0.31309	0.30271	**0.32109**	0.30207
Traffic	0.69392	**0.99571**	0.99567	0.99561	0.99540	0.99530	0.00827	**0.27759**	0.27676	0.27272	0.26893	0.26402
PeopleOnStreet	0.68385	0.99491	0.99495	0.99534	0.99536	**0.99569**	0.00970	0.28711	**0.28768**	0.27931	0.28062	0.28500

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2924910, IEEE Transactions on Circuits and Systems for Video Technology

(a) (b) (c) (d) (e) (f)

Fig. 8. Key sensitivity analysis of the #1 decoded frame in Johnny and Traffic sequences (a) Johnny original frame (b) Johnny encrypted frame (c) Johnny decrypted frame with error key (d) Traffic original frame (e) Traffic encrypted frame (f) Traffic decrypted frame with error key

6) Analysis of NPCR and UACI

The number of pixels changing rate (*NPCR*) and unified average change intensity (*UACI*) [3, 41] are widely used to test the robustness of image cryptosystems against the differential cryptanalysis [3]. They are calculated by

$$NPCR = \frac{\sum_{u,v} O(u,v)}{h \times w} \times 100\%, \tag{33}$$

$$O(u,v) = \begin{cases} 0 & if \ C_1(u,v) = C_2(u,v) \\ 1 & if \ C_1(u,v) \neq C_2(u,v) \end{cases}, \tag{34}$$

$$UACI = \frac{1}{h \times w}\left[\sum_{u,v} \frac{C_1(u,v) - C_2(u,v)}{2^L - 1}\right] \times 100\%, \tag{35}$$

where h and w are the height and width of a frame, respectively. $C_1(u, v)$ and $C_2(u, v)$ represent the pixel values of two cipher frames at the position (u, v), respectively. Then, the expected value of *NPCR* and *UACI* are computed as

$$NPCR_{Expected} = \left(1 - 2^{-L}\right) \times 100\%, \tag{36}$$

$$UACI_{Expected} = \frac{1}{2^{2L}} \frac{\sum_{u=1}^{2^L - 1} u(u+1)}{2^L - 1} \times 100\% \cdot \tag{37}$$

As the length of the pixel value L is 8 bits, $NPCR_{Expected}$ and $UACI_{Expected}$ are 99.6094% and 33.4635%, respectively. The closer the values of *NPCR* and *UACI* to $NPCR_{Expected}$ and $UACI_{Expected}$ are, the better the encryption performance is. Experiments are performed to test *NPCR* and *UACI* of 12 video sequences as [23] by using five encryption algorithms, and the results are listed in Table VIII. It can be found that Sen can achieve the optimal *NPCR* and *UACI* in 6 and 5 video sequence, respectively. So Sen outperforms the other algorithms in *NPCR* and *UACI*. Nevertheless, the performance of Enc in *NPCR* is not so satisfactory, but its performance in *UACI* is better than other 3 algorithms.

7) Analysis of Key Sensitivity

In general, a good encryption algorithm should be sensitive to the initial key. It means that the encrypted video is still significantly distorted after decryption if a slight change is made to the key. In the experiment, the initial key is defined as {0x01, 0x09, 0x09, 0x01, 0x01, 0x02, 0x02, 0x08, 0x01, 0x09, 0x09, 0x04, 0x00, 0x03, 0x01, 0x06}, and the key in decryption process is set as {0x00, 0x09, 0x09, 0x01, 0x01, 0x02, 0x02, 0x08, 0x01, 0x09, 0x09, 0x04, 0x00, 0x03, 0x01, 0x06}. Some experimental results are shown in Fig.8, where the encryption algorithm is Sen and QP=24. From Fig.8, it can be found that the #1 decrypted frames of Johnny and Traffic sequence have similar distortion as the encrypted frames when the initial key only has a small difference of one bit. Therefore, an illegal user cannot obtain any useful information from the encrypted video when the key is unknown to him.

TABLE IX
THE AVERAGE ENCRYPTION SPACE OF A SINGLE GOP IN FIVE ENCRYPTION ALGORITHMS

Video	QP	Algorithm				
		Glenn [14]	Sallam [23]	Boyajis [26]	Enc	Sen
Mobile	8	288160	399711	408280	420276	670049
	24	69113	85793	91249	98087	163673
	40	11690	11945	15129	16234	26877
Foreman	8	200920	234439	243136	251127	333314
	24	19470	21363	25468	28442	39194
	40	1783	1926	2890	3398	4542
BQSquare	8	247416	319008	327554	336382	478500
	24	41211	50642	56289	60132	91818
	40	6602	6682	9440	9908	15634
Basket-ballPass	8	118752	148903	156381	162952	223547
	24	21257	23147	25586	27982	42822
	40	1811	1859	2694	3207	4538
BQMall	8	1127505	1296096	1335708	1369545	1908068
	24	98560	110270	123706	133542	209054
	40	10865	11208	15913	17993	26172
Par-tyScene	8	1265477	1646292	1685618	1721579	2558657
	24	230706	265143	288561	305836	494225
	40	22780	23148	32337	35103	53595
Johnny	8	1348344	1453344	1529922	1576207	1906032
	24	53313	58845	67579	72493	105883
	40	6034	6257	9478	10669	14652
FourPeo-ple	8	1381081	1502048	1581943	1639071	2139641
	24	83224	93785	109661	115938	175484
	40	13050	13390	19176	20552	30489
Basket-ballDrive	8	4783429	5552148	5747621	5842386	6730446
	24	208110	225826	248473	268004	370465
	40	18389	19336	25455	30993	41529
Kimono	8	5093368	5470980	5681116	5784675	8627746
	24	229469	275059	283145	302080	532762
	40	25988	27715	30844	35818	59957
Traffic	8	7921558	8900578	9217222	9523708	1.34×10^7
	24	632658	701882	782504	844394	1315269
	40	77877	80779	103101	110918	172924
People-OnStreet	8	8516297	1.02×10^7	1.05×10^7	1.08×10^7	1.45×10^7
	24	909240	1036419	1144257	1276675	1863448
	40	88203	97624	136658	169194	229312

8) Analysis of Encryption Space

The encryption space represents the number of the encrypted syntax elements. Generally, an encryption algorithm needs to guarantee sufficient encryption space to achieve good security. Table IX lists the encryption space of 5 encryption algorithms in 12 sequences with different QPs. It can be found that the increase of QP significantly reduces the encryption space of all algorithms. The reason is that the residual has more zero coefficients when QP is large. Since the zero coefficients cannot be encrypted, the performance of all encryption algorithms will be reduced. This also has been proved in the objective evaluation results in Section V.*B.2)*. Compared with other encryption algorithms designed in [14, 23, 26], Enc and Sen have relatively larger encryption space. Especially for Sen, it greatly increases the encryption space because of the additional scrambling to the residual coefficients. Even the QP is large, Sen can still guarantee sufficient encryption space.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2924910, IEEE Transactions on Circuits and Systems for Video Technology

TABLE X
THE AVERAGE BIT RATE INCREMENT OF THREE ENCRYPTION ALGORITHMS

Video	Algorithm		
	Boyajis [26]	Enc	Sen
Mobile	0.0177	0.0244	0.0737
Foreman	0.0135	0.0261	0.0884
BQSquare	0.0240	0.0302	0.0765
BasketballPass	0.0263	0.0430	0.0954
BQMall	0.0216	0.0320	0.0892
PartyScene	0.0174	0.0233	0.0695
Johnny	0.0200	0.0345	0.1044
FourPeople	0.0246	0.0348	0.1078
BasketballDrive	0.0119	0.0290	0.0779
Kimono	0.0038	0.0192	0.0981
Traffic	0.0164	0.0249	0.0986
PeopleOnStreet	0.0151	0.0292	0.0979

9) Analysis of Bit Rate Change

In a video SE algorithm, the bit rate change of the encrypted video is an important index. In general, the ideal situation is to keep the video bit rate after encryption. However, from the analysis in Section V.*B.1)*, as only the syntax elements encoded in bypass mode are chosen for encryption in Glenn's scheme [14] and Sallam's scheme [23], they can keep the bit rate, but their visual distortion is obviously insufficient compared with the other three SE algorithms. The significant information of the Johnny sequence can even be observed from the encrypted frames. Therefore, this paper tries to improve the security with an acceptable bit rate increment. The average bit rate increment for each video of three SE algorithms is listed in Table X, and it can be found that the bit rate is inevitably increased as long as the syntax elements encoded in regular mode are encrypted. The bit rate increment of Enc (almost under 4%) is slightly higher than that of Boyajis's scheme [26], because more syntax elements in regular mode such as chroma IPM is encrypted. However, it can provide better encryption performance. Furthermore, due to additional coefficient scrambling, the bit rate of Sen has an average increment of 8.978%, However, Sen can provide the largest distortion and edge loss.

VI. DISCUSSIONS

Based on the performance comparison results between Enc and Sen in Section V, it can be found that coefficient scrambling can significantly distort the videos with some bit rate increments. Here, the necessity of encrypting chroma IPM and the effect of encrypting luma IPM on chroma IPM are discussed.

According to Section IV.*A.2)*, there are five chroma IPMs (planar, vertical, horizontal, DC, and corresponding luma IPM) in H.265/HEVC, and their corresponding number are 0, 26, 10, 1, and 36, respectively. If the corresponding luma IPM is one of the first four modes, the mode which is the same as the corresponding luma IPM is changed to 34, and then the list of the first four modes is changed according to the luma IPM. They are presented in Table XI. Here, the probabilities of no change in chroma IPM after encrypting luma IPM are analyzed. Assuming that the probability of no change in chroma IPM after encrypting luma IPM is $p(A)$, there exists two situations.

Situation 1. the chroma IPM is the corresponding luma IPM.

In this situation, the chroma IPM equals to the luma IPM and it is changed by the luma IPM. As there are 35 luma IPMs and the encryption is pseudo-random, so it can obtain $p(A)=1/35$.

TABLE XI
CHROMA IPM NUMBER

Luma IPM number	Chroma IPM number			
0 (planar)	34	26	10	1
26 (vertical)	0	34	10	1
10 (horizontal)	0	26	34	1
1 (DC)	0	26	10	34

Situation 2. the chroma IPM is one of the first four modes: planar, vertical, horizontal and DC.

The probability of no change in chroma IPM is more complex in this situation and it can be further classified to two cases.

Case 1. the luma IPM is one of the first four modes: planar, vertical, horizontal and DC.

From Table XI, each luma IPM is corresponded to four chroma IPMs. Here, when a luma IPM is encrypted, the probability of no change in chroma IPM is calculated as follows.

Assuming the original luma IPM is planar, the encrypted luma IPM can be no change (event B_1), one of 26, 10, and 1(event B_2), and one of the other 31 modes (event B_3). It can obtain that $p(B_1)=1/35$, $p(B_2)=3/35$ and $p(B_3)=31/35$. According to Table XI, it can get $p(A|B_1)=1$, $p(A|B_2)=1/2$ and $p(A|B_3)=3/4$. Based on the Bayes' theorem, one has

$$p(A) = p(A \mid B_1)\, p(B_1) + p(A \mid B_2)\, p(B_2) + p(A \mid B_3)\, p(B_3). \tag{38}$$

Thus, $p(A)=103/140$. With the same way, the same probability can be obtained with a different original luma IPM.

Case 2. the luma IPM is one of the other 31 modes.

Assuming the original luma IPM is 2, the encrypted luma IPM has two possibilities, and they are one of 0, 26, 10 and 1 (event C_1) and one of the other 31 modes (event C_2). It can get $p(A|C_1)=3/4$ and $p(A|C_2)=1$ according to Table XI. Based on the Bayes' theorem,

$$p(A) = p(A \mid C_1)\, p(C_1) + p(A \mid C_2)\, p(C_2). \tag{39}$$

Thus, one has $p(A)=136/140$. With the same way, the same probability can be obtained with a different original luma IPM.

From the above analysis, it can be found that the probability of no change in chroma IPM in Situation 1 is 2.86%. Therefore, it is no need to encrypt chroma IPM in Situation 1. However, in Situation 2, the probabilities of no change in chroma IPM of two Cases are up to 73.57% and 97.14%, respectively. So encrypting chroma IPM in Situation 2 is necessary and can further distort the chroma components. To verify this point, some experiments are conducted to compare the performance of only encrypting luma IPM (Luma) and encrypting both chroma IPM and luma IPM (Luma+Chroma), and the corresponding results are listed in Table XII. The experimental setup is the same as that in Section V, where *PSNR*(U) represents the *PSNR* of U component and *PSNR*(V) represents that of V component. Observing Table XII, one can see 63 results are best in terms of 72 *PSNR*s when chroma IPM and luma IPM are both encrypted. Encrypting both chroma IPM and luma IPM can further reduce *PSNR*(U) and *PSNR*(V) by up to 3.96 dB and 3.66 dB, respectively, but the bit rate is only increased by the maximum of 0.0027. However, there are also situations that both bit rate increment and *PSNR* are reduced after further encrypting chroma IPM for BQSquare, FourPeople, BasketballDrive and Kimono

TABLE XII
THE COMPARISON RESULTS OF LUMA ENCRYPTION AND LUMA+CHROMA ENCRYPTION WITH DIFFERENT QP

Video	QP	*PSNR*(U)		*PSNR*(V)		Bitrate increment	
		Luma	Luma+ Chroma	Luma	Luma+ Chroma	Luma	Luma+ Chroma
Mobile	8	19.04	17.59	17.20	14.72	0.0079	0.0097
	24	19.54	16.34	17.75	15.42	0.0195	0.0221
	40	18.01	16.72	15.53	15.89	0.0258	0.0275
Foreman	8	27.45	26.43	22.90	23.03	0.0081	0.0098
	24	26.96	26.13	24.53	24.27	0.0187	0.0189
	40	28.26	26.62	20.16	21.29	0.0136	0.0138
BQSquare	8	28.25	26.09	24.16	22.13	0.0064	0.0079
	24	29.77	25.81	22.46	20.15	0.0294	0.0301
	40	31.84	31.83	25.27	23.50	0.0362	0.0355
Basket-ballPass	8	25.74	22.80	22.78	21.43	0.0160	0.0187
	24	24.10	24.90	23.42	21.59	0.0400	0.0417
	40	24.58	24.91	21.82	22.34	0.0228	0.0232
BQMall	8	26.28	24.56	27.23	24.65	0.0065	0.0078
	24	24.64	24.41	25.40	24.77	0.0357	0.0371
	40	25.31	24.36	24.70	24.46	0.0224	0.0228
Par-tyScene	8	24.47	22.44	23.26	20.74	0.0057	0.0071
	24	24.37	22.01	23.01	20.78	0.0238	0.0256
	40	24.35	23.20	23.40	22.28	0.0225	0.0227
Johnny	8	24.20	22.88	26.48	24.67	0.0132	0.0148
	24	22.98	23.32	25.64	25.48	0.0279	0.0284
	40	22.29	22.30	25.35	24.74	0.0188	0.0189
FourPeo-ple	8	25.99	25.05	28.41	28.21	0.0136	0.0153
	24	26.10	25.50	28.15	27.33	0.0378	0.0380
	40	26.10	24.82	27.70	27.62	0.0223	0.0222
Basket-ballDrive	8	23.74	22.62	22.58	21.91	0.0092	0.0111
	24	22.96	22.78	23.14	21.56	0.0143	0.0146
	40	23.38	23.10	21.48	21.21	0.0124	0.0123
Kimono	8	22.56	22.54	21.66	22.62	0.0063	0.0086
	24	28.06	25.40	29.68	26.36	0.0020	0.0020
	40	29.22	28.28	32.08	30.77	0.0030	0.0029
Traffic	8	21.62	20.01	24.32	21.92	0.0096	0.0117
	24	21.28	20.51	23.72	20.06	0.0247	0.0260
	40	23.55	22.68	26.93	25.16	0.0148	0.0149
People-OnStreet	8	26.55	25.05	27.58	24.98	0.0100	0.0119
	24	27.05	25.58	27.85	25.94	0.0187	0.0190
	40	26.36	25.39	27.24	26.24	0.0167	0.0168

with QP=40. Thus, it indicated that the encryption of chroma IPM is essential and can further distort videos.

VII. CONCLUSIONS

In this paper, a tunable H.265/HEVC selective encryption scheme is proposed based on the analysis of the relationship between H.265/HEVC syntax elements and encryption performance. It is composed of two parts Enc and Scr. In Enc, multiple syntax elements are chosen for encryption, especially the encryption of chroma IPM can introduce larger distortion. Scr is an optional scrambling to enhance the protection of edge information. Experimental results and analysis show that Enc can enhance the distortion of the video at low bit rate increment, and Sen can further increase the encryption space and produce large edge loss. The proposed scheme is compatible to the standard H.265/HEVC codec, and provides better protection in video content and edge region compared with the existing SE algorithms. Enc and Sen can be implemented for different scenarios in practical applications. Reducing the bit rate increment and improving the encryption distortion deserve further investigation.

ACKNOWLEDGEMENT

The authors would like to thank for the anonymous reviewers for their kind comments and suggestions for improving the paper, and thank for my colleague Professor Cheng-qing Li for his contribution on the improvement of English writing.

REFERENCES

[1] R. Iqbal, S. Shirmohammadi, A. E. Saddik, and J. Zhao, "Compressed-domain video processing for adaptation, encryption, and authentication," *IEEE Multimedia*, vol. 15, no. 2, pp. 38–50, 2008.

[2] C. Li, D. Lin, J. Lü, F. Hao, "Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography," *IEEE MultiMedia*, vol. 25, no. 4, pp. 46-56, 2018.

[3] C. Li, D. Lin, B. Feng, J. Lü, F. Hao, "Cryptanalysis of a chaotic image encryption algorithm based on information entropy," *IEEE Access*, vol. 6, pp. 75834-75842, 2018.

[4] M. Long, F. Peng, and H.-y. Li, "Separable reversible data hiding and encryption for HEVC video," *J. Real-Time Image Process.*, vol. 14, no. 1, pp. 171–182, 2018.

[5] L. Qiao, K. Nahrstedt, "A new algorithm for MPEG video encryption," *in Proc. Int. Conf. Imaging Sci. Syst. Technol.*, Las Vegas, Nevada, USA, 1997, pp. 21-29.

[6] J. Shah and V. Saxena, "Video encryption: A survey," *Int. J. Comput. Sci. Iss.*, vol. 8, no. 2, pp. 525–533, 2011.

[7] Grangetto, E. Magli, and G. Olmo, "Multimedia selective encryption by means of randomized arithmetic coding," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 905–917, 2006.

[8] M. N. Asghar and M. Ghanbari, "An efficient security system for CABAC bin-strings of H.264/SVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 3, pp. 425–437, 2013.

[9] Y. Wang, M. O'Neill, and F. Kurugollu, "A tunable encryption scheme and analysis of fast selective encryption for CAVLC and CABAC in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 9, pp. 1476–1490, 2013.

[10] F. Peng, X.-w. Zhu, and M. Long, "An ROI privacy protection scheme for H.264 video based on FMO and chaos," *IEEE Trans. Inf. Forensics and Security*, vol. 8, no. 10, pp. 1688–1699, 2013.

[11] F. Peng, X.-q. Gong, M. Long, and X.-m. Sun, "A selective encryption scheme for protecting H. 264/AVC video in multimedia social network," *Multimedia Tools Appl.*, vol. 76, no. 3, pp. 3235-3253, 2017.

[12] G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand et al., "Overview of the high efficiency video coding(HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[13] G. Van Wallendael, J. De Cock, S. Van Leuven, A. Boho, P. Lambert, B. Preneel, and R. Van de Walle, "Format-compliant encryption techniques for high efficiency video coding," *in Proc. IEEE Int. Conf. Image Process.*, Melbourne, Australia, 2013, pp. 4583–4587.

[14] G. Van Wallendael, A. Boho, J. De Cock, A. Munteanu, and R. Van de Walle, "Encryption for high efficiency video coding with video adaptation capabilities," *IEEE Trans. Consumer Electron.*, vol. 59, no. 3, pp. 634–642, 2013.

[15] H. Hofbauer, A. Uhl, and A. Unterweger, "Transparent encryption for HEVC using bit-stream-based selective coefficient sign encryption," *in Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Florence, Italy, 2014, pp. 2005–2009.

[16] Z. Shahid and W. Puech, "Visual protection of HEVC video by selective encryption of CABAC binstrings," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 24–36, 2013.

[17] M. Farajallah, W. Hamidouche, O. Deforges, and S. E. Assad, "ROI encryption for the HEVC coded video contents," *in Proc. IEEE Int. Conf. Image Process.*, Quebec, Canada, 2015, pp. 3096–3100.

[18] Y. Tew, K. Minemura, S. K. Wong, "HEVC selective encryption using transform skip signal and sign bin," i*n Proc. Asia-Pac. Signal and Inf. Process. Assoc. Annu. Summit Conf.*, Honolulu, Hawaii, USA, 2016, pp. 963–970.

[19] F. Peng, H.-y. Li, and M. Long, "An effective selective encryption scheme for HEVC based on rossler chaotic system," *in Proc. Int. Symp. Nonlin. Theor. Its Appl.*, Hong Kong, China, 2015, pp. 1–4.

[20] M. Yang, L. Zhuo, J. Zhang, and X. Li, "An efficient format compliant video encryption scheme for HEVC bitstream," *in Proc. IEEE Int. Conf. Inform. Comput.*, Nanjing, China, 2015, pp. 374–378.
[21] V. A. Memos and K. E. Psannis, "Encryption algorithm for efficient transmission of HEVC media," *J. Real-Time Image Process.*, vol. 12, no. 2, pp. 473–482, 2016.
[22] A. I. Sallam, E.-S. M. El-Rabaie, and O. S. Faragallah, "Efficient HEVC selective stream encryption using chaotic logistic map," *Multimedia Syst.*, vol. 24, no. 4, pp. 419–437, 2018.
[23] A. I. Sallam, O. S. Faragallah, and E.-S. M. El-Rabaie, "HEVC selective encryption using RC6 block cipher technique," *IEEE Trans. Multimedia*, vol. 20, no. 7, pp. 1636–1644, 2018.
[24] A. I. Sallam, E.-S. M. El-Rabaie, and O. S. Faragallah, "CABAC- based selective encryption for HEVC using RC6 in different operation modes," *Multimedia Tools Appl.*, vol. 77, no. 21, pp. 28395–28416, 2018.
[25] J. Li, C. Wang, X. Chen, Z. Tang, G. Hui, and C.-C. Chang, "A selective encryption scheme of CABAC based on video context in high efficiency video coding," *Multimedia Tools Appl.*, vol. 77, no. 10, pp. 12837–12851, 2018.
[26] B. Boyadjis, C. Bergeron, B. Pesquet-Popescu, and F. Dufaux, "Extended selective encryption of H.264/AVC(CABAC)-and HEVC-encoded video streams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 4, pp. 892–906, 2017.
[27] W. F. Ehrsam, S. M. Matyas, C. H. Meyer, and W. L. Tuchman, "A cryptographic key management scheme for implementing the data encryption standard," *IBM Syst. J.*, vol. 17, no. 2, pp. 106–125, 1978.
[28] Sklavos N, Koufopavlou O, "Architectures and VLSI implementations of the AES-proposal Rijndael," *IEEE Trans. comput.*, vol. 51, no. 12, pp. 1454-1459, 2002.
[29] R. Zimmermann, A. Curiger, H. Bonnenberg, H. Kaeslin, N. Felber, and W. Fichtner, "A 177 Mb/s VLSI implementation of the international data encryption algorithm," *IEEE J. Solid-St. Circuits*, vol. 29, no. 3, pp. 303–307, 1994.
[30] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.
[31] V. Sze and M. Budagavi, "A comparison of CABAC throughput for H.265/HEVC vs. H.264/AVC," *in Proc. IEEE Workshop Signal Process. Syst.*, Taipei, Taiwan, 2013, pp. 165–170.
[32] H. Lipmaa, P. Rogaway, and D. Wagner, "Ctr mode encryption," *in Proc. NIST, Comput. Security Resource Center, First Modes Operat.Workshop*. [Online]. Available: http://csrc.nist.gov/CryptoToolkit/modes/workshop1/papers/lipmaa-ctr.pdf, 2000.
[33] C. M. Fu, E. Alshina, A. Alshin, Y. W. Huang, C. Y. Chen, C. Y. Tsai, C. W. Hsu, S. M. Lei, J. H. Park, and W. J. Han, "Sample adaptive offset in the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1755–1764, 2012.
[34] S. Gupta and S. G. Mazumdar, "Sobel edge detection algorithm," *Int. J. Comput. Sci. Manag. Res.*, vol. 2, no. 2, pp. 1578–1583, 2013.
[35] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., man, cybern.*, vol. 9, no. 1, pp. 62–66, 1979.
[36] S. Winkler and P. Mohandas, "The evolution of video quality measurement: from PSNR to hybrid metrics," *IEEE Trans. on Broadcast.*, vol. 54, no. 3, pp. 660–668, 2008.
[37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.
[38] H. Isa, I. Bahari, H. Sufian, and M. R. Z'aba, "AES: Current security and efficiency analysis of its alternatives," *Proc. 7th Int. Conf. Inf. Assurance Secur. (IAS)*, Malacca, Malaysia, 2011, pp. 267–274.
[39] H. Ahmed, H. Kalash, and O. Allah, "Encryption efficiency analysis and security evaluation of RC6 block cipher for digital images," *in Proc. IEEE Int. Conf. Electr. Eng.*, Hong Kong, China, 2007, pp. 1-7.
[40] J. Ahmad and F. Ahmed, "Efficiency analysis and security evaluation of image encryption schemes," *Proc. Int. J. Video Image Process. Netw. Secur.*, vol. 12, no. 4, pp. 18–31, 2012.
[41] V Patidar, N K Pareek, G Purohit, et al, "A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption," *Opt. Commun.*, vol. 284, no. 19, pp. 4331-4339, 2011.

Fei Peng (M'17) received the Ph.D. degree in Circuits and Systems from the South China University of Science and Technology, Guangzhou, China, in 2006. He was a visiting fellow of the Department of Computer Science, University of Warwick, U.K. in 2009–2010. He was a visiting professor in SeSaMe Centre, College of Computing, National University of Singapore in 2016. Currently, he is a professor in the College of Computer Science and Electronic Engineering, Hunan University, Changsha. His areas of interest include digital watermarking and digital forensics.

Xiang Zhang received the M.S. degree in Computer Science and Technology from Jiangxi University of Science and Technology, Ganzhou, Jiangxi, China, in 2016. Currently, He is a Ph.D. candidate of the College of Computer Science and Electronic Engineering, Hunan University, Changsha. His areas of interest include multimedia security and coverless image steganography.

Zi-Xing Lin received the B.S. degree in Information Security from Hunan University, Changsha, Hunan, China, in 2014. Currently, He is a Ph.D. candidate of the School of Computer Science and Electronic Engineering, Hunan University, Changsha. His areas of interest include multimedia security and digital watermark.

Min Long received the Ph.D. degree in Circuits and Systems from the South China University of Science and Technology, Guangzhou, China, in 2006. She was a visiting fellow of the Department of Computer Science, University of Warwick, U.K. in 2009–2010. Currently, she is a professor in the College of Computer and Communication. Her areas of interest include digital watermarking and chaos-based secure Communication.